

Técnicas Avanzadas de Middleware

Primera aplicación distribuida



Objetivo de la Práctica

- Introducir el concepto de Middleware.
Construcción de un sistema distribuido con poco esfuerzo. Analizar las alternativas.
- Transparencia del middleware. Revisar los conceptos básicos.
- Revisar la tipología de middleware y situar la práctica dentro de la taxonomía. En particular, analizar las implicaciones de otras arquitecturas (P2P).
- Establecer preguntas sobre conceptos habituales.

Concepto de Middleware

- Introducir el concepto de Middleware.
 - Revisar las definiciones de middleware
 - Razonar si Remoting es un middleware tradicional o con características modernas
 - Es transparente para el programador.

Transparencia

- EL middleware intenta ser transparente:
 - Los alumnos prueban diferentes elementos del lenguaje. Muchos parecen funcionar sin tocar el código.
 - Se observan numerosos casos de transparencia.

Lenguaje de programación

- Algunas características del lenguaje afectan al middleware:
 - Representación de los datos (big little endian)
 - Tipos de datos (enteros, flotantes, nullables, etc.)
 - Diferencias entre lenguajes afecta al diseño. Por ejemplo, ¿qué pasa si Java no tiene ushort?
 - Programación funcional
 - Tipado dinámico
 - Extensión de tipos dinámicos

Arquitectura

- Analizar con los alumnos los elementos de la práctica que podrían cambiar al alterar la filosofía del middleware:
 - Nombrado
 - Descubrimiento
 - Descripción del funcionamiento
 - Mecanismos de interacción

Preguntas

- **Naming** in distributed systems:
 - *How does the client find the server?*
 - *How a name can be resolved in the network?*
 - *Can a service be discovered by looking for its properties?*
 - *How many address may be assigned?*
 - *What abstractions and services are available to do this?*

Preguntas

- **Remote Objects:**
 - *Who create the remote (local) object?*
 - *How long does it live?*
 - *How does a component learn about another's interface?*
 - *What services does it provide?*

Preguntas

- Remote **communication modes**:
 - *What is a remote invocation?*
 - *How can an integer be transferred over the network?*
 - *Does the client wait for a response from the server?*
 - *What if there is an error on the network?*
 - *How is programmed?*
 - *What are the limitations of remote invocation, are there any at all?*
 - *How are messages transferred from one component to another?*
 - *How do we encode service parameters, return values in these messages?*
 - *How do the components at either end know which encoding to use?*

Actividades

- Dividir la clase en varios grupos de trabajo
- Cada grupo debe decidir si la **transparencia** de los conceptos de programación tienen implicaciones en el diseño del middleware. Deben decidir y razonar si:
 - La implicación es muy grave, importante o leve
 - Si es algo particular para un middleware o de un lenguaje de programación
- En estos momentos del curso no se evalúan estos comentarios. Solo se apuntan para futuras discusiones.